EXHIBIT B

Amendments to the Claims

1.   (Currently amended) A system for distributing an application environment ~~to a compute node~~ comprising:

a compute node comprising a computer system;

a first storage unit for storing blocks of a root image of the compute node, wherein the first storage unit comprises a first non-volatile memory, wherein the root image comprises a computer program, wherein the blocks comprise sections of data, and wherein a file of the root image comprises at least one block;

a second storage unit for storing a leaf image, the leaf image comprising new data blocks and changes to the blocks of the root image, wherein the second storage unit comprises a second non-volatile memory; and

a union block device for interfacing between the compute node and the first and second storage units to distribute the application environment to the compute node, wherein the union block device comprises a driver, wherein the union block device creates the application environment by merging the blocks of the root image stored on the first storage unit with the blocks of the leaf image stored on the second storage unit.

2.   (Original) The system as recited in Claim 1 wherein the compute node comprises a server.

3.   (Original) The system as recited in Claim 1 wherein the compute node comprises a thin-client workstation.

4.   (Original) The system as recited in Claim 1 wherein the root image comprises an operating system.

5.   (Original) The system as recited in Claim 1 wherein the root image is concurrently accessible to a plurality of compute nodes.

.

6.   (Original) The system as recited in Claim 1 wherein the first storage unit is remotely located from the compute node.

7.   (Original) The system as recited in Claim 1 wherein the second storage unit is remotely located from the compute node.

8.   (Original) The system as recited in Claim 1 wherein the second storage unit contains a block modification log for the compute node.

9.   (Original) The system as recited in Claim 1 wherein the first storage unit in contained within a first partition on a hard disk and the second storage unit is contained within a second partition on the hard disk.

10.   (Original) The system as recited in Claim 1 wherein the union block device comprises a low-level driver for interfacing between the first and second storage units and the file system of the compute node.

11.   (Original) The system as recited in Claim 1 wherein the union block device, upon receiving a write request from the compute node for a sector X, creates an appropriate persistent mapping for sector X.

12.   (Original) The system as recited in Claim 11 wherein the union block device writes sector X on the second storage unit.

13.   (Original) The system as recited in Claim 1 wherein the system operates in a high performance computing cluster.

14.   (Original) The system as recited in Claim 1 wherein the system operates in a grid computing cluster.

15.   (Original) The system as recited in Claim 1 wherein the first storage unit is read only.

16.   (Original) The system as recited in Claim 1 further comprising an intermediate image between the root image and the leaf image, wherein the intermediate image comprises an application environment service group.

17.   (Currently amended) A method for distributing an application environment ~~to a compute node~~ comprising:

storing blocks of a root image of a[[the]] compute node on a first storage unit, wherein the compute node comprises a computer system, and wherein the first storage unit comprises a first non-volatile memory, wherein the root image comprises a computer program, wherein the blocks comprise sections of data, and wherein a file of the root image comprises at least one block;

storing a leaf image comprising new data blocks and changes to the blocks of the root image on a second storage unit, wherein the second storage unit comprises a second non-volatile memory;

merging the blocks of the root image stored on the first storage unit with the blocks of the leaf image stored on the second storage unit to create the application environment; and

delivering the application environment to the compute node.

18.   (Original) The method as recited in Claim 17 further comprising:

modifying the leaf image in response to the compute node's access to the application environment.

19.   (Original) The method as recited in Claim 18 wherein the modifying comprises:

upon receiving a write request from the compute node for a sector X, creating an appropriate persistent mapping for sector X; and

writing sector X on the second storage unit.

20.   (Original) The method as recited in Claim 17 further comprising:

reconciling the root image and the leaf image to form a new root image.

21.   (Original) The method as recited in Claim 17 wherein the compute node comprises a server.

22.   (Original) The method as recited in Claim 17 wherein the compute node comprises a thin-client workstation.

23.   (Original) The method as recited in Claim 17 wherein the root image comprises an operating system.

24.   (Original) The method as recited in Claim 17 wherein the root image is concurrently accessible to a plurality of compute nodes.

.

25.   (Original) The method as recited in Claim 17 wherein the first storage unit is remotely located from the compute node.

26.   (Original) The method as recited in Claim 17 wherein the second storage unit is remotely located from the compute node.

27.   (Original) The method as recited in Claim 17 wherein the second storage unit contains a block modification log for the compute node.

28.   (Original) The method as recited in Claim 17 wherein the first storage unit is contained within a first partition on a hard disk and the second storage unit is contained within a second partition on the hard disk.

29.   (Original) The method as recited in Claim 17 wherein merging occurs at an operational level between the first and second storage units and file system of the compute node.

30.   (Original) The method as recited in Claim 17 wherein the method operates in a high performance computing cluster.

31.   (Original) The method as recited in Claim 17 wherein the system operates in a grid computing cluster.

32.   (Original) The method as recited in Claim 17 wherein the first storage unit is read only.

33.   (Original) The method as recited in Claim 17 further comprising:

creating an intermediate image on a third storage unit between the root image and the leaf image, wherein the intermediate image comprises an application environment service group.